Support for ECI over TCP/IP in CICS TS V2.2 or later

Darren Beard

CICS Developer

darren_beard@uk.ibm.com

4121B

impact·venture*

This presentation contains details of ECI support over TCP/IP.

IMPACT IBM

What is ECI?

- **E**xternal **C**all **I**nterface
  - An API which allows a non-CICS program running on a client to call a CICS program located on a CICS server.

IMPACT IBM

ECI allows a non-CICS client program to link to a CICS program. This enables the client to make use of existing server routines that could be used, for example, to make enquiries on a database.

ECI - External Call Interface

Types of ECI call

- A client program can make the following types of call to a server:
  - Program link calls, which may be synchronous or asynchronous.
  - Calls to retrieve responses from a previous asynchronous call.
  - Calls that return a value indicating the status of the CICS region.

Program link calls can be either synchronous or asynchronous. Synchronous calls mean that the calling program waits for a response from the linked to program. Asynchronous calls allow the two programs (client and server programs) to continue independently.

ECI - External Call Interface

What is EPI?

- **E**xternal **P**resentation **I**nterface
  - An API which allows a non-CICS client program to appear to a CICS server as one or more standard 3270 terminals.
  - The EPI consists of a set of calls which can be made from a client program.

Use of the EPI allows clients to access CICS programs which were written for 3270 terminals without the need to change the CICS application code.

The EPI consists of a set of calls which may be made from a client:-

Initialize the EPI
Terminate the EPI
Attach a virtual terminal
Detach a virtual terminal
etc.

EPI - External Presentation Interface

SNA supported clients

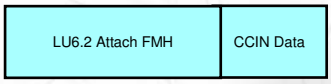
- Support for CICS clients was added in version 4.1
  - Supports ECI and EPI.
  - LU6.2 connections required.

Initial support for clients was shipped in CICS/ESA 4.1. This support was for both ECI and EPI. The only communications protocol into CICS supported was LU6.2.

ECI - External Call Interface
EPI - External Presentation Interface

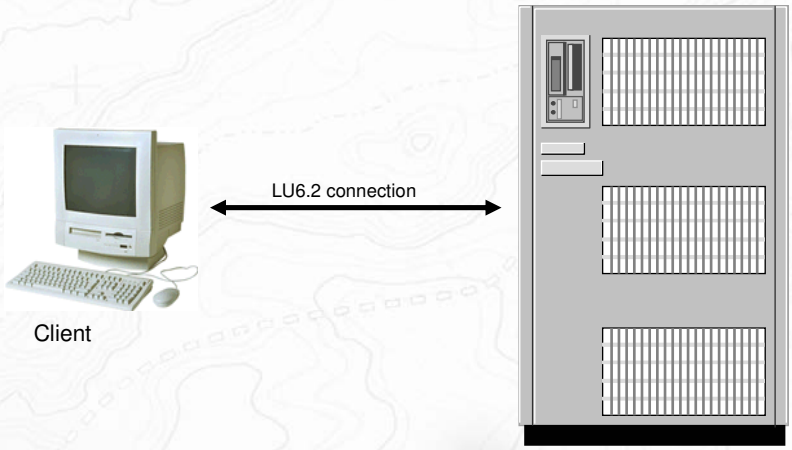
Running ECI work

- Clients install by running transaction CCIN for function INSTALL.
- ECI flows to CICS are then accepted.
- When client is finished, it runs CCIN for function UNINSTALL.



The CCIN data contains information about the client eg. it's codepage, client code version, client capabilities etc. It is an architected header and parameters. This follows an LU6.2 attach FMH. For a client install or uninstall request, the FMH will attach transaction CCIN, which is the architected transaction for client installation and deletion.

SNA support



SNA support has the client machine connected via an LU6.2 connection (single or parallel session, sync level 1) to CICS.

IMPACT IBM

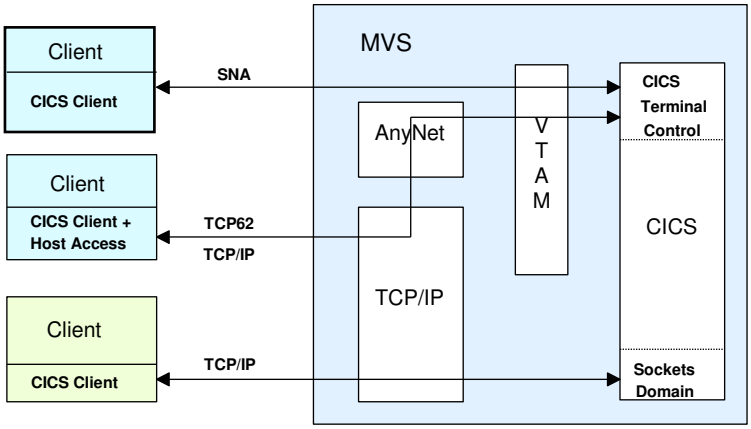
Problems with SNA support

- SNA setup is often quite difficult.
- If TCP/IP network being used, other products such as TCP62 are required for protocol conversion.
  - This has setup and performance implications.

IMPACT IBM

If clients were to be used in a network containing TCP/IP, then it was necessary to have protocol conversion taking place between TCP/IP and SNA. This meant that it was necessary to understand both SNA and TCP/IP, so setting up the networks was sometimes a problem. Also, the fact that protocol conversion was needed on every conversation flow meant that performance was degraded.

ECI Connection Types



Customers using SNA can connect their clients straight into CICS using LU6.2 and VTAM into CICS terminal control. To use TCP/IP, the client communicates with the Anynet feature of MVS which performs protocol conversion and then communicates with VTAM and CICS terminal control. The overhead of this on every conversation flow is obviously not desirable. Also, the configuration of this scenario is not easy. It requires a knowledge of both VTAM and TCP/IP.

For TCP/IP, it would be nice if we did not have to do protocol conversion but could instead communicate directly with CICS using TCP/IP. Since it is CICS sockets domain which handles TCP/IP, then the obvious solution is to allow ECI flows to come in to CICS through sockets domain and not to use terminal control at all.

The advantage of connecting directly to Sockets Domain is that there is no requirement for Anynet or even VTAM. Therefore, the configuration steps for all these components are eliminated. Datastream conversion does not occur, reducing overhead, and access to CICS TS is opened up to any platform which supports TCP/IP.

TCP/IP Support for ECI

- A solution to the SNA problems is to provide ECI support over TCP/IP. This was shipped in CICS TS V2.2.
- The following items were required of the design
  - Must not require client upgrades.
  - Must not require changes to either client or server application programs.
  - Performance must be at least as good as SNA supported clients.
  - Must be able to support large numbers of clients.

Two of the requirements of the design were that it must not require client upgrades and that it must not be necessary to change either client or server application programs.

NB. There is a pre-req of using at least CICS Universal Client version 3. Moreover, clients prior to 3.13 do not support conversation level ping and some timeout situations properly. As a result, it is recommended that clients are at least at the 3.13 level.

With regard to performance, the aim was to be at least as good as a client using an LU6.2 connection with the SNA support.

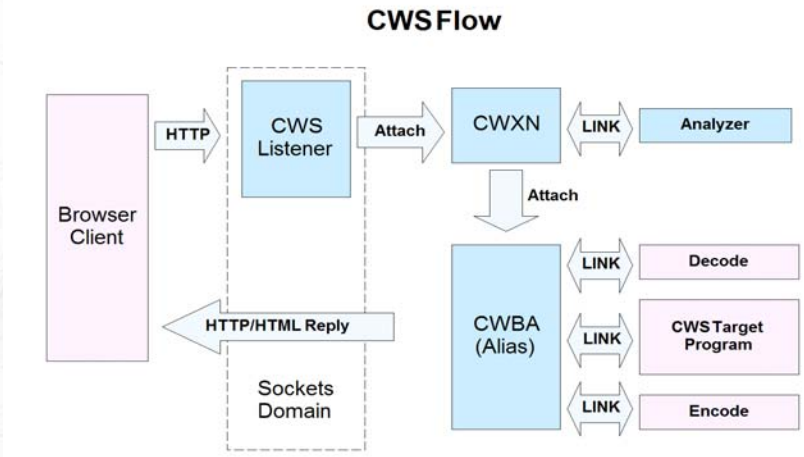
It was also necessary to be able to support a large number (several thousands potentially) of clients simultaneously. This requirement necessitated a number of changes to sockets domain.

Sockets Domain

- Provides the interface between TCP/IP and CICS.
- Listens for input on predefined ports.
- Upon receipt of input on a predefined port, starts a transaction which is associated with the port.

The sockets domain takes care of TCP/IP communications to and from CICS regions. It provides a domain interface for use by CICS for sending and receiving data over TCP/IP ports.

Sockets domain listens for input on ports defined via TCPIPSERVICE definitions. When input arrives on one of these ports, sockets domain initiates a transaction which is also specified on the TCPIPSERVICE definition.



The foil shows CICS Web Support processing for a Web browser request to run a program on CICS, using CICS Web Support. The steps are:

- The browser enters a URL indicating the host, port, target program and optionally data for that program.

- The CICS listener for the port attaches the CWXN transaction, which LINKs to a URM known as the 'analyser' to determine what program the user wishes to execute and make other CWS-specific choices.

- CWXN then attaches a CICS-supplied transaction called the 'alias' to process the request. The default name for the alias transaction is CWBA, but any name can be used. The alias is a kind of mirror program tailored especially for the web. It uses Web Domain services to interface to Sockets Domain for receiving inputs and sending back outputs.

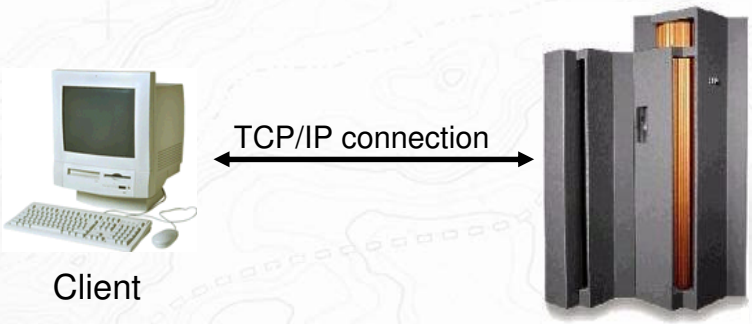
- The alias first calls an optional, user-supplied 'converter' program to decode the Web-format input data into the COMMAREA format which the target program expects.

- The alias then LINKs to the target program. Note that the target program can reside in a different CICS region from the alias, but the alias must be in the same region as the listener.

- When the target program returns control, the alias LINKs to the converter again, this time for it to convert the COMMAREA produced by the program into HTML for the browser.

- The alias sends the results, using Web Domain services, which in turn use Sockets Domain services, to the client.

ECI Support CICS TS V2.2



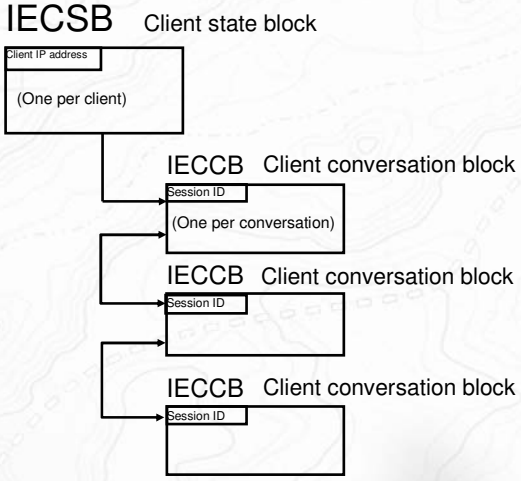
From CICS TS V2.2 onwards, clients are able to use TCP/IP to connect directly to CICS. The client code and the server code will not be aware of which communications protocol is being used by a client.

Implementation

- A new CICS domain, the IE domain, has been introduced to provide ECI specific services and also
  - New trace levels IE = 1 or 2
  - New dump parameters IE = 1, 2 or 3
- A new TCPIPSERVICE definition is supplied.
- TCPIPSERVICE definition extensions
  - New PROTOCOL value - ECI
  - New option ATTACHSEC - LOCAL or VERIFY
- A new CICS supplied transaction, CIEP.
- Mirror restructure.
- Architecture consistent with CWS and IIOP support.

To assist with diagnosing any problems with the IE domain, IE level tracing has been added to the CETR transaction. A new dump formatting keyword of IE has been added and this supports values of 1, 2 or 3. This will format out the IE domain control blocks.

IE domain control blocks



This foil shows the main IE domain control blocks used in handling an ECI request from a TCP/IP attached client. It is IE domain which keeps track of the client conversations and the associated states of those conversations. We avoid the need to have sockets permanently associated with specific clients by using the control blocks in IE domain.

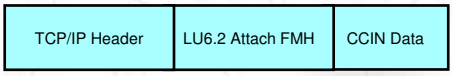
IECSBs are chained from the IE domain anchor block. There is one IECSB per client. A number of IECCBs will be chained from the IECSB. There is one IECCB per conversation. These control blocks are formatted out in the IE section of a CICS system dump.

IECSB - Client state block. This performs roughly the same function as a TCTSE (system entry) does for SNA attached clients.

IECCB - Client conversation block. This performs roughly the same function as a TCTTE (session entry) does for SNA attached clients.

Running ECI work over TCP/IP (1)

- Client installs by running CCIN INSTALL as for SNA.
- Based on information in the TCP/IP header:
  - IE domain builds an IECSB for the client.
  - Stores information relating to the client.
- All this processing is done under CIEP.



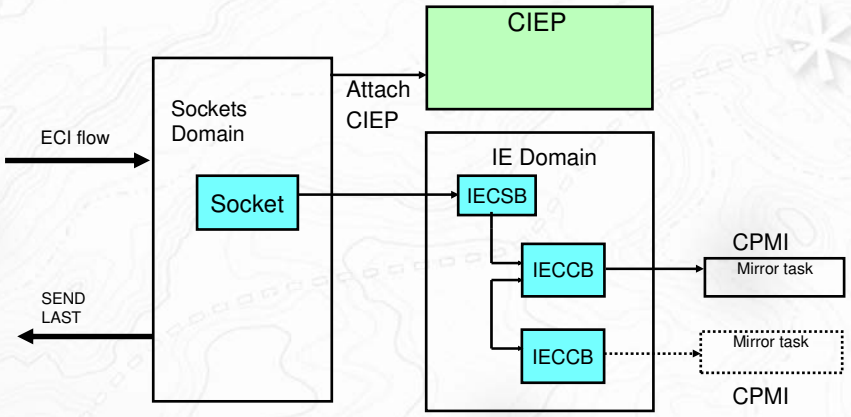
A standard TCP/IP header precedes the FMH and CCIN data when a client is attached via a TCP/IP connection. The header contains important information such as a session identifier and information relating to the connection status. This header data may be followed by other 'request dependent' data as part of the overall TCP/IP header.

Running ECI work over TCP/IP (2)

- Run ECI workload.
  - IE domain builds an IECCB for each active client conversation.
- Client uninstalls by running CCIN UNINSTALL. Again, it is CIEP which performs this processing.

The application programs are all run under the mirror task. The CCIN UNINSTALL processing is all performed under the CIEP task. For TCP/IP attached clients, the CCIN task is never run. All the equivalent processing is performed by CIEP.

Overall view of events for a non-extended conversation

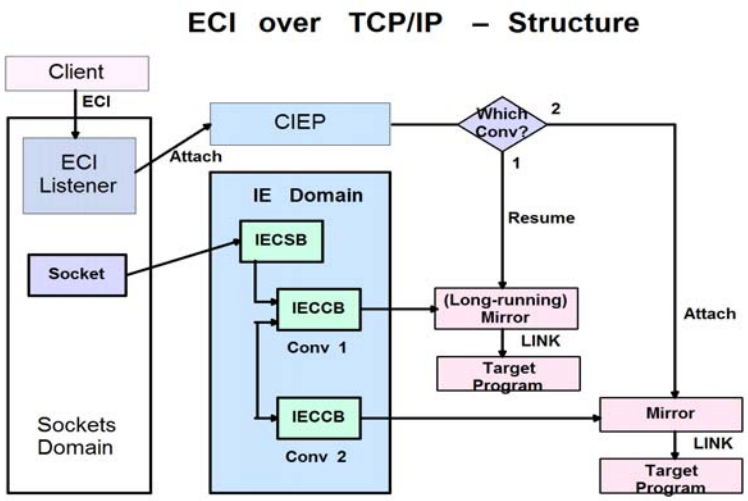


The foil shows the structure of the control blocks used in handling an ECI request from a client. The Socket object in the sockets domain points to an IECSB which belongs to the CIEP transaction. A chain of IECCBs is hung from the IECSB. For conversations which are active, the relevant IECCB will have an associated mirror task.

FLOWS

Each inbound ECI flow will cause a CIEP task to be attached by sockets domain. CIEP issues an XMAT attach for the mirror with a new facility type indicating that this is an ECI request.

DFHXMAT invokes a new callback module, DFHIEXM, as part of the attach processing of the mirror. The INIT stage will sign on the client to the USER domain, using the userid and password supplied (if they are required by the TCPIPSERVICE in use for this client). The only function in the BIND routine is to collect monitoring and statistics information.

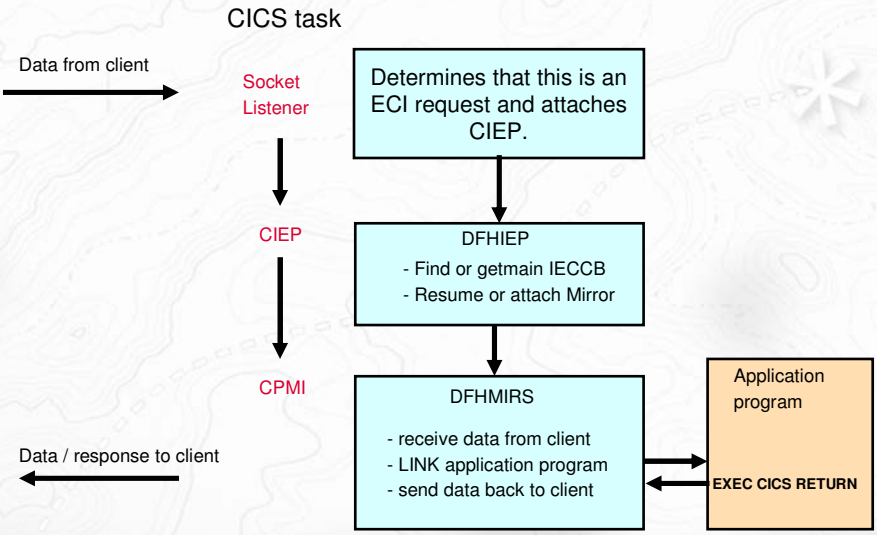


This foil shows the structure of the main control blocks used in managing ECI requests from a client. The socket control block for this client (in the Sockets Domain) points to a “client state block” (IECSB) representing the client in the IE Domain. Chained off the IECSB are two IECCS “client conversation” control blocks, each of which represents one *conversation* (session) between the client and the CICS region. While it probably will be most common to have a single conversation in progress at any one time, a client can have multiple sessions concurrently, as might occur in a client using several applications from different windows, or if the client is a gateway. The client in the foil happens to have two independent conversations in progress.

For each ECI request, Sockets domain attached transaction CIEP. CIEP determines whether the request is for a new or for an existing conversation from the ECI input parameters. For a new conversation it creates a new IECCB control block and attaches a new mirror. The mirror will become long-running for an “extended” ECI request (conversation 1 here) and terminate on completion of the ECI request. For an existing conversation, CIEP finds the corresponding IECCB and resumes the mirror transaction to which it points.

The mirror task gets attached (XMAT attach) with a new primary client (IP_ECI) and receives the IECCB as the primary client request block. The program, DFHMIRS, has been modified to use the DFHIEIE SEND and RECEIVE functions of the IE domain rather than Terminal Control (DFHTC) requests if the primary client is IP_ECI.

Task View



This foil attempts to show the tasks which will be seen running in CICS whilst various parts of the processing are occurring.

Initially the socket listener task, typically task number 00003, receives input from a client. The only function of the listener is to attach the correct transaction, use the ephemeral port for further communications and clear the listening port of activity. For an ECI flow, the sockets domain will attach transaction CIEP.

The function of CIEP is to see whether this client is known or not. If the client is known (ie. an IECSB exists for it) then is the conversation known (ie. does an IECCB exist for it). If there is no IECCB, one is getmained. If an IECCB already exists, then a suspended mirror associated with it is resumed. If an IECCB does not already exist then there is no suspended mirror to resume, so a new mirror is attached.

The mirror (CPMI) then handles the requests in the usual manner, except that it uses IEIE send and receive instead of TC send and receive. The mirror will link to the application program requested and return the data and responses directly to the client without the need to invoke CIEP again.

CONVERSATION STATE

- Conversation state tracking simpler than for LU6.2
- Bracket state
  - Existence of IECCB means 'in bracket'
- Contention state not needed because no bidding
- Chain state
  - All flows 'only in chain' and CD or CEB
  - ieccb_session_state is SEND or RECEIVE
- User state
  - No allocate or syncpoint status
  - ieccb_user_state is SEND or RECEIVE

Conversation state

The use of brackets and change direction indicators means that there is a need to keep track of conversation state in some way comparable to the LU6.2 state machines. Fortunately, it works out much simpler for IE than for LU6.2.

The bracket state machine is not required because the existence of an IECCB for a particular session id means that it is in-bracket. No IECCB means between-brackets.

The contention state machine is not required because there is no bidding. CICS never initiates requests to the client.

The chain state machine is much simpler because all flows are OIC and CD (or CEB). All that is required is an indication of whether the session is in SEND or RECEIVE state. This is held in ieccb_session_state. When an IECCB is created, the session is set to SEND state because it was the receipt of data that caused the IECCB to come into existence and the next valid socket request must be a send. The session is switched to RECEIVE state immediately following SO SEND in the mirror task. For an extended conversation, the session switches back to SEND state immediately after the data received have been matched to an existing IECCB.

The user state machine is much simpler because there is no need to keep any allocate or syncpoint status. All that is required is an indication of whether the user (mirror program) is in SEND or RECEIVE state. This is held in ieccb_user_state. When an IECCB is created, the user is set to RECEIVE state because the mirror task will always start with a receive. The user is switched to SEND state during the completion of an IEIE RECEIVE request. For an extended conversation, the user switches back to RECEIVE state during the completion of an IEIE SEND request.

READ TIMEOUT

- The CICS architected PING flows are used to handle READ TIMEOUT
  - connection status 02 in TCP/IP header is ping request
  - connection status 04 in TCP/IP header is ping response
  - following data indicates for CONVERSATION ping what the session id and sequence number are
- CONVERSATION level PING support is indicated on CCIN INSTALL flow.
- All clients must support CONNECTION level PING.
- What happens when a timeout occurs depends on whether CONVERSATION level PING is supported or not.
- NOTE: The supplied definition for default mirror CPMI has RTIMOUT(NO).

The timeout value used on the WAIT_MVS is extracted from the mirror transaction PROFILE RTIMOUT value (PPFTRTO) by DFHIEXM at transaction initiation and stored in IECCB_TIME_OUT. As is the case for this value in the PROFILE, zero means no time out. PPFTRTO is a bin(8) with each unit representing 16.78 seconds.

NB. The standard mirror CPMI is defined with profile DFHCICSA, which has RTIMOUT(NO). This means that long running mirrors will wait indefinitely for data unless the user customises the RTIMOUT value for the mirror transaction.

CONVERSATION PING

- CONVERSATION ping request is sent after first time out, containing session id and conversation sequence number.

- CONVERSATION ping reply received from a client has three flavours
  - ►NOT KNOWN

    Expected data probably crossed PING in transit. Abend mirror if still waiting, else do nothing. Conversation probably already terminated.
  - ►ABENDED

    Client was in send state so may be hung. Chop the conversation without further flows.
  - ►NOT ABENDED

    Client was in receive state so its outbound data may have crossed our PING request in transit. Check if data has arrived. If not, abend the conversation.

- If receive for CONVERSATION PING times out then initiate CONNECTION PING processing.

Clients indicate whether they support conversation level time out when they install into CICS. If the TIMEOUT variable length parameter is not present, it is assumed that conversation level time out is not supported. When a conversation level PING is received by CICS from a client, one of three possible PING replies will be sent and the corresponding actions performed.

- If the conversation is in SEND state (the mirror has not yet replied) the 'abend conversation' reply will be sent and the mirror task (IECCB) marked for abend.

- If the conversation is in RECEIVE state the assumption is that the mirror's reply must have crossed the PING request. A PING reply of 'not abended' is sent and no further action is taken.

- If the conversation is not known, a PING reply of 'not known' is sent.

If conversation level PING is supported by the client, a conversation level PING request will be sent to the client by a mirror task whose IEIE RECEIVE has timed out. The time out interval will be taken from the RTIMOUT value in the mirror transaction's PROFILE. The mirror task will be abended if the receive for the PING reply also times out.

Race conditions

PING race conditions can arise if network delays result in both ends sending a PING request. These will be resolved according to the architecture, with the conversation in SEND state making the decision to abend the conversation.

Repeated time out

The time out of a conversation level PING request could be assumed to mean that the client is no longer in a position to respond to any flow for any conversation. To verify this, the architecture requires that a connection level PING should be sent at this point. The client will only be uninstalled if this request also times out.

CONNECTION PING

- CONNECTION PING request sent when receive for CONVERSATION PING has timed out, or on data receive time out if client does not support CONVERSATION PING.
- There are no variable values in the CONNECTION PING request or reply.
- If CONNECTION PING reply received, a scan is done for timed-out IECCBs. These are marked to abend and then posted.
- If CONNECTION PING reply times out, the client is uninstalled.

Connection level ping

This is part of the base CICS TCP/IP architecture.

When a connection level PING request is received, all that is required is to send the PING reply to indicate that we are still alive. There are no variable values in the PING reply.

If conversation level PING is not supported by the client, a connection level PING request will be sent to the client by a mirror task whose IEIE RECEIVE has timed out. The time out interval will be taken from the RTIMOUT value in the mirror transaction's PROFILE. If the receive for the PING reply also times out, the mirror task will be abended, the client uninstalled and the socket closed.

CTIN ATTACH

- EPI over native TCP/IP is not supported.
- CTIN is the install transaction for EPI.
- CIEP explicitly checks for CTIN as this is likely to be a common error.
- Error returned is a standard CTIN response CTIN_INSTALL_CANCELLED.

EPI is not supported via TCP/IP. A method of implementing an equivalent to EPI would be to use the CICS bridge facility to access 3270 applications on CICS from a client machine. This can be implemented using the ECI bridge support also available in CICS TS V2.2. The bridge support in turn relies on the native TCP/IP support for ECI.

The error returned to the client if CTIN is attempted over TCP/IP is a pre-existing response so that the use of this support does not necessitate any specific client upgrades.

TCPIPSERVICE (1)

```
VIEW TCPIPS(ECI) GR(DFH£SOT)
OBJECT CHARACTERISTICS                                  CICS RELEASE = 0620
 CEDA  View TCpipservice( ECI      )
  TCpipservice    : ECI
  GROup           : DFH£SOT
  DEscription     : ECI TCPIPSERVICE
  Urm             :
  POrtnumber      : 01435              1-65535
  STatus          : Open               Open | Closed
  PRotocol        : Eci                Iiop | Http | Eci
  TRansaction     : CIEP
  Backlog         : 00005              0-32767
  TSqprefix       :
  Ipaddress       :
  SOcketclose     : No                 No | 0-240000 (HHMMSS)
 SECURITY
  SSl             : No                 Yes | No | Clientauth
  Certificate     :
  AUthenticate    : No                 No | Basic | Certificate | AUTORegister
+                                      | AUTOMatic

                                                    SYSID=CIST APPLID=IYK2ZAF2

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```


This foil shows the new TCPIPSERVICE definition which is supplied in group DFH£SOT. This group is not in DFHLIST. The default (well known) port number for ECI requests is 1435. This number can be changed by supplying an alternative TCPIPSERVICE definition. The TCPIPSERVICE definition in the foil also shows transaction CIEP as the transaction to be initiated as a result of input being received.

TCPIPSERVICE (2)

```
 VIEW TCPIPSERVICE(ECI) GR(DFH£SOT)
 OBJECT CHARACTERISTICS                                  CICS RELEASE = 0620
  CEDA  View TCpipservice( ECI      )
+  ATtachsec       : Verify            Local | Verify
   DNS CONNECTION BALANCING
    DNsgroup       : CICS
    GRPcritical    : No                No | Yes
                                                  SYSID=CIST APPLID=IYK2ZAF2
PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```


The Attachsec parameter is new on TCPIPSERVICE for CICS TS V2.2. It is necessary so that we can enforce a security requirement on the client. The default setting for Attachsec is VERIFY. Clients may be coming in from anywhere and so are not trusted by default.

NB. The attachsec setting is only interrogated for ECI type TCPIPSERVICEs. It has no effect on HTTP or IIOP.

New SIT parameter

- A new SIT parameter, MAXSOCKETS has been introduced in CICS TS V2.1. It is the ECI support which really necessitated this parameter.

A new SIT parameter has been introduced to enable control of the maximum number of sockets which may be managed by Sockets Domain in a given CICS region. The parameter is called MAXSOCKETS. If the userid under which the CICS job is running has superuser authority, then the default value is 65535. If not, the default value is the value of the MAXFILEPROC parameter specified in SYS1.PARMLIB member BPXPRMxx.

NB. Sockets created by Java programs running on threads that are not managed by CICS do not count towards the limit.

CEMT showing socket data

```
I TCPIP
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tcp Ope Act(00005) Max( 00100 )

                                                SYSID=CICS APPLID=IYK2ZAF2
RESPONSE: NORMAL                              TIME:  16.14.57  DATE: 21.05.07
  PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```


ACTSOCKETS shows how many sockets are currently in use on the CICS region. It would not normally be higher than MAXSOCKETS. It may be higher temporarily if MAXSOCKETS has just been reduced and the work has not had chance to drain.

MAXSOCKETS specifies how many sockets the CICS region is allowed to use.

The value of MAXSOCKETS can be changed through the CEMT command. The foil shows the result of issuing CEMT I TCPIP.

Attempt to set MAXSOCKETS too high

```
SET TCPIP MAX(2000)
STATUS:  RESULTS - OVERTYPE TO MODIFY
Tcp Ope Act(00001) Max( 01500 )                          EXCEEDS HARDLIMIT









                                                SYSID=CICS APPLID=IYK2ZAF2
RESPONSE: 1 ERROR                             TIME:  16.18.11  DATE: 21.05.07
  PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```


An attempt was made to set MAXSOCKETS to 2000 when the value in SYS1.PARMLIB was set to 1500. The CICS job did not have superuser authority. The CEMT screen displays the message "Exceeds hardlimit" and CICS sets the value to 1500.

Setting MAXSOCKETS too high as a SIT override results in a new message (DFHSO0124) being issued during CICS initialization and MAXSOCKETS being set to the hard limit value.

SO Domain Stats Record

- Current MAXSOCKETS value
- Current and peak number of inbound sockets (these are in CTS 1.3)
- Current and peak number of outbound sockets
- Current and peak number of persistent outbound sockets
- Number of outbound sockets created
- Number of inbound sockets created
- Number of outbound sockets closed
- Number of inbound sockets closed

The introduction of the MAXSOCKETS parameter and the sockets domain restructure has necessitated the introduction of some additional statistics information. It is now possible for the creation of sockets to be queued because the current number is at the maximum. There are also some new fields specifically for the outbound socket support.

This foil and the next show the new SO domain statistics record fields.

SO Domain Stats Record (continued)

- Number of times at MAXSOCKETS
- Number of create sockets delayed at MAXSOCKETS
- Total time that create sockets were delayed at MAXSOCKETS
- Current and peak number of create sockets delayed at MAXSOCKETS
- Number of socket creates that timed out at MAXSOCKETS
- Waiting time of currently delayed create requests

These statistics are likely to be useful in dealing with ECI support throughput problems.

Message DFHSO0126 is issued to CSMT when a connect request is received whilst CICS is at MAXSOCKETS.

Monitoring

- Number of persistent socket create requests
- Number of non-persistent socket create requests
- Non-persistent socket count
- Persistent socket count
- Number of socket receive requests
- Number of characters received

For monitoring, for each transaction, the data indicated on this foil and the next will be monitored.

Monitoring (continued)

- Number of socket send requests
- Number of characters sent
- Total number of socket requests
- Total outbound socket i/o wait time

No notes for this foil.

Migration (CICS)

- CICS TS V2.2
  - Create TCPIPSERVICE definition(s)
  - Ensure that TCPIP=YES is set in the SIT
  - Set MAXSOCKETS to an appropriate value
  - Install
    - TCPIPSERVICE definition(s)
    - TD queue CIEO (group DFHDCTG)
    - Transaction CIEP (group DFHIPECI)
    - Program DFHIEP (group DFHIPECI)

If different clients are to use different ports or have different ATTACHSEC settings, then you need to create TCPIPSERVICE definitions in CICS for these new values. The definitions must be installed into the CICS region with which the clients will communicate.

The groups, DFHDCTG and DFHIPECI, containing the standard supplied definitions, are included within DFHLIST so they are available by default.

Migration (Clients)

- Client Workstation
  - Edit client initialization file to specify
    - Communications protocol of TCP/IP
    - Port number in TCPIPSERVICE definition
    - TCP/IP name of CICS TS System/390 server
  - Optional - uninstall TCP62

For older clients, the initialization file is CICSCLI.INI. For more recent versions for the client and gateway, the initialization file is CTG.INI. The TCP/IP section needs to specify the same port number as is defined on the corresponding CICS TCPIPSERVICE definition. A section from an INI file is shown below. This specifies the NETNAME of a CICS region, specifies PROTOCOL as TCPIP and PORT as 1435.

```
SECTION SERVER = CICSTCPX
   DESCRIPTION=TCP/IP Server
   UPPERCASESECURITY=N
   USENPI=N
   PROTOCOL=TCPIP
   NETNAME=winmvs2c.hursley.ibm.com
   PORT=1435
   CONNECTTIMEOUT=0
   TCPKEEPALIVE=N
ENDSECTION
```

If any other products, such as TCP62, have been used to support SNA communications which are no longer needed, they can now be uninstalled from the client machines.

IMPACT IBM

Summary

- ECI over TCP/IP results in
  - Simpler configuration
  - Reduced administration costs
  - No intermediate products
  - No SNA - TCP/IP conversion
  - Access to CICS on platforms without SNA or TCP62
  - Easy migration path
  - No changes required to either client or server programs
  - Performance and manageability enhancements

IMPACT IBM

The foil lists some of the benefits of using native TCP/IP support for ECI applications.

More information

- CICS TS V2.2 Infocenter
  - look up the CIEP transaction
  - look in the Interproduct Communication book
    - Chapter 3 CICS Clients gives a lot of information
- IBM Redbook
  - *CICS Transaction Gateway V5 The WebSphere Connector for CICS (SG24-6133-01).*
    - Chapter 5 covers a lot of information relevant to ECI support in CICS.

The foil lists some of the places where you can find more information on this topic should you wish to do so.

IBM 2007 IMPACT

IBM

Questions
and
Answers

impact·venture*

IBM 2007 IMPACT

IBM

The end.